2

NASA CR-134461

NASA-CR-134461) NASIS DATA BASE MANAGEMENT SYSTEM: IBM 360 TSS IMPLEMENTATION. VOLUME 7: OPERATING (Neoterics, Inc., Cleveland, Ohio.) ~~41~~ 42 p HC $4.25

N73-30145

CSCL 09B G3/08

Unclas 13483

NASIS DATA BASE MANAGEMENT SYSTEM - IBM 360 TSS IMPLEMENTATION

VII - OPERATING SPECIFICATIONS

NEOTERICS, INC.

prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NASA Lewis Research Center
Contract NAS 3-14979

41

TABLE OF CONTENTS

I. INTRODUCTION

This section of the documentation concerns itself with the environment which surrounded the design, implementation and installation of NASIS. It describes how the various components of NASIS fit together and how the system is to be installed and modified.

It is assumed that any individual involved in the modification or installation of NASIS has a complete working knowledge of TSS as well as PL/I.

II. ENVIRONMENT

NASIS is a data management system which runs under TSS/360 on the IBM System 360 Model 67.

A. TSS/360

TSS/360 is a programming system that gives multiple simultaneous user access to the computer facilities. The full system is explained in other documents which are published by IBM. Some of these are:

'Concepts and Facilities,' C28-2003

'Terminal User's Guide,' GC28-2017

'Command System User's Guide,' C28-2001

'System Messages,' GC28-2037

'PL/I Programmer's Guide,' GC28-2049

'PL/I Language Reference Manual,' GC28-2045.

The above list is by no means comprehensive. It merely indicates the most commonly used of the available manuals.

B. Standard TSS-ID Profile

The current mode of operation uses four prime TSS-IDs which control and operate NASIS. Each of these TSS-IDs serves a fundamental and important role in the operation of NASIS. These TSS-IDs are SALISR**, SADBA***, SACWNER*, and SAMTT***.

SALISR's responsibility is to contain and maintain the source and object code of all NASIS modules. It also maintains the libraries from which NASIS is executed as well as other supporting data sets.

In other words, the entire physical entity known as NASIS is maintained by SALISR. The person responsible for maintaining SALISR is responsible for the integrity of NASIS.

SADBA is the Data Base Administrator's TSS-ID. This ID contains a library for all user written modules, such as file loading and formatting routines. This library is also used to contain all messages and explanations pertaining to specific data bases. The data set NASIS.USERIDS, which contains a list of all JOINed NASISIDs, is also maintained here.

SAOWNER is the owner of all the files which are to be used with NASIS. This TSS-ID is maintained by the Data Base Administrator. The integrity and the use of all data bases is the responsibility of the DBA (Data Base Administrator).

SAMTT is the TSS-ID under which operation in MT/T mode is possible. There is only one such TSS-ID. This is practical because the one MT/T TSS-ID will permit many people to use NASIS concurrently, and also because this ID must have a special privilege class because of system limitations.

Any number of other TSS-IDs can and do exist and each of these can run NASIS in stand-alone mode.

In order to create a common environment, compatable with NASIS, for the various TSS-IDs which may exist, two executable data sets have been created: THE.USERID.CREATION and THE.USERID.ANALYSIS. These data sets are owned by SALISR. Therefore, for any new TSS-ID on which this environment is desired, the following commands must be executed:

```
_SHARE  THE.USERID,SALISR,THE.USERID
_BACK   THE.USERID.CREATION
  (wait for completion)

  (if you wish to review your TSS-ID)
_EXECUTE THE.USERID.ANALYSIS
```

Please refer to Appendix A for the details of THE.USERID.ANALYSIS and Appendix B for the details of THE.USERID.CREATION.

C. NASIS System Structure

Because of the increasing size and complexity, as well as the prospect for distribution of the NASIS system, it was decided to structure the NASIS data sets. This structuring was approached with the functions of system backup and restore in mind. The structure is explained below and is currently in effect.

A private VAM disk pack (LABEL=SAFETY) has been incorporated as the foundation for this structure. This disk pack will function as the on-line residence volume for NASIS. It will contain everything necessary to bring up NASIS at any other TSS installation. The backup copies of all actively used NASIS data sets, as well as the master copies of all non-actively used NASIS data sets, from SALISR, SADBA, and SAMTT will be on this volume. The data sets currently defined for residence on SAFETY are:

```
LINK.NASISLIB (BACKUP)
NASIS.NASYSLIB (BACKUP)
NASIS.MTTLIB  (BACKUP) (FROM: SAMTT)
NASIS.DBALIB (BACKUP) (FROM: SADBA)
NASIS.USERJOIN  (BACKUP)
NASIS.SALISR.USERLIB (BACKUP)
NASIS.SAMTT.USERLIB (BACKUP) (FROM: SAMTT)
NASIS.SOURCE.module  (MASTER)
NASIS.SOURCE.LISRMAC  (BACKUP)
LINK.SOURCE.module (MASTER)
NASIS.RSOURCE  (BACKUP)
NASIS.RINDEX  (BACKUP)
NASIS.UTILITYS  (BACKUP) (FROM:SAUTY)
NASIS.ANALYSIS  (BACKUP)
NASIS.CREATION  (BACKUP)
NASIS.USERIDS  (BACKUP) (FROM: SADBA)
```

This structure also provides a simplified, efficient, flexible method of NASIS backup and restore, i.e., to use the TSS command DMPRST to dump or restore the entire disk pack.

None of the data sets on disk pack SAFETY are PERMITted; therefore, they cannot be accessed by anyone but their owner, TSS userid SALISR. The data sets necessary for the use and maintenance of NASIS and its files are all maintained in public storage under TSS userid SALISR. Data sets USERJOIN, and NASISLIB are PERMITted to all users with read-only access. These are the only data sets necessary for the normal use of NASIS.

The remaining data sets, NASYSLIB, RSOURCE,

RINDEX, THE.USERID.CREATION, THE USERID.ANALYSIS, and all SOURCE data sets (including SOURCE.LISRMAC) are PERMITted to the system implementors with read-only access so that they may modify and test the system (i.e., these data sets should be SHAREd from SALISR).

The contents and functions of the NASIS data sets are as follows:

NASISLIB(0) - the job library which contains the LINKEDITed stand-alone version of NASIS and the NASIS message file. Upon thorough testing, this library is copied to SAMTT as the new MTT system.

LINK.NASISLIB - the backup copy of NASISLIB(0).

NASYSLIB(0) - the job library which contains the non-LINKEDITed stand-alone version of the NASIS and the NASIS system message file. This data set should be used only by system implementors for testing new or modified modules.

NASIS.NASYSLIB - the backup copy of NASYSLIB(0).

NASIS.MTTLIB - the backup copy of MTTLIB(0). From SAMTT.

NASIS.DBALIB - the backup copy of DBALIB(0). From SADBA.

USERJOIN - the input command stream for creating the appropriate BUILTINs, PROCDEFs and messages necessary to allow a TSS userid to function as a NASIS data base user. It also provides shared access to the necessary job libraries and PERMITted data bases.

NASIS.USERJOIN - the backup copy of USERJOIN.

NASIS.SALISR.USERLIB - the backup copy of the USERLIB from TSS-ID salisr. This library contains all the procdefs necessary to maintain NASIS and SALISR.

NASIS.SAMTT.USERLIB - the backup copy of the USERLIB from TSS-ID SAMTT. This library contains all the procdefs necessary to run NASIS in MTT mode and maintain SAMTT.

SOURCE.module - the temporary copy of the source

code for a NASIS module. It should be shared and copied by system implementors to serve as a basis for modifications. (These datasets only exist while actually being modified.)

NASIS.SOURCE.module - the master copy of the source code for a NASIS module.

SOURCE.LISRMAC - the working copy of the PL/I macro library, containing all of the preprocessor components, as well as any other standard sections of coding included into any of the NASIS modules.

NASIS.SOURCE.LISRMAC - the backup copy of the NASIS PL/I macro library.

LINK.SOURCE.module - the master copy of the source code for the modules required for linkediting.

RSOURCE - the working copy of the source portion of the NASIS assembler language macro library.

NASIS.RSOURCE - the backup copy of RSOURCE.

RINDEX - the working copy of the index to the NASIS assembler language macro library (RSOURCE).

NASIS.RINDEX - the backup copy of RINDEX.

NASIS.UTILITYS - the backup copy of the UTILITYS. From SAUTY.

THE.USERID.ANALYSIS - the working copy of the data set which is executed to yield the current TSS status of a given TSS-ID.

NASIS.ANALYSIS - the backup copy of THE.USERID.ANALYSIS.

THE.USERID.CREATION - the working copy of the data set which is executed to create a standard userid.

NASIS.CREATION - the backup copy of THE.USERID.CREATION.

NASIS.USERIDS - the backup copy of NASIS.USERIDS. From SADBA.

D. NASIS System Procedures

This section lists the procedures used for the maintenance of NASIS:

1. LINKALL

```
 PROCDEF LINKALL
DISPLAY ' SUPPORT SYSTEM'
LINKSUPP
DISPLAY ' RETRIEVAL SYSTEM'
LINKRET
DISPLAY ' MAINTENANCE SYSTEM'
LINKMAIN
DISPLAY ' PRINT MONITOR'
LINKPRNT
DISPLAY ' REST OF THE SYSTEM'
LINKREST
```

2. LINKMAIN

```
 PROCDEF LINKMAIN
DEFAULT LINKWMSG=N
VV LINK.SOURCE.MAINTAN0,SOURCE.MAINTAN0
VV LINK.SOURCE.MAINTAIN,SOURCE.MAINTAIN
DDEF NASYSLIB,VP,NASYSLIB(0),,,,,,OLD
DDEF MAINTAN0,VP,MAINTAN0
ERASE MAINTAN0
DDEF MAINTAN0,VP,MAINTAN0,,,,,,NEW,,TLU
DISPLAY ' INCLUDING'
INK MAINTAN0,Y,MAINTAN0,,N,N,N
REL NASYSLIB
DDEF NASISLIB,VP,NASISLIB(0),,,,,,OLD
ERASE NASISLIB(0)(MAINTAIN)
DISPLAY ' COMBINING'
LNK MAINTAIN,Y,NASISLIB,,N,Y,Y
REI MAINTAN0,NASISLIB
ERASE MAINTAN0
FR LIST.MAINTAIN(0),,,EDIT,Y
ERASE SOURCE.MAINTAN0
ERASE SOURCE.MAINTAIN
```

3. LINKOUT

```
 PROCDEF LINKOUT
PARAM $01,$02
RELEASE BACKUPDD
IF '$02'  = '';ERASE LINK.SOURCE.$01;-
DDEF BACKUPDD,VI,LINK.SOURCE.$01,DISP=NEW,-
UNIT=(DA,2314),VOLUME=(,SAFETY);-
VV SOURCE.$01,LINK.SOURCE.$01
IF '$02' ¬= '';ERASE LINK.$01;-
DDEF BACKUPDD,$02,LINK.$01,DISP=NEW,-
```

```
UNIT=(DA,2314),VOLUME=(,SAFETY);-
VV $01(0),LINK.$01
RELEASE BACKUPDD
```

4. LINKPRNT

```
 PROCDEF LINKPRNT
DEFAULT LINKWMSG=N
VV LINK.SOURCE.PRINTS0,SOURCE.PRINTS0
VV LINK.SOURCE.PRINTMON,SOURCE.PRINTMON
DDEF NASYSLIB,VP,NASYSLIB(0),,,,,,OLD
DDEF PRINTS0,VP,PRINTS0
ERASE PRINTS0
DDEF PRINTS0,VP,PRINTS0,,,,,,NEW,,TLU
DISPLAY ' INCLUDING'
INK PRINTS0,Y,PRINTS0,,N,N,N
REL NASYSLIB
DDEF NASISLIB,VP,NASISLIB(0),,,,,,OLD
ERASE NASISLIB(0)(PRINTMON)
DISPLAY ' COMBINING'
LNK PRINTMON,Y,NASISLIB,,N,Y,Y
REI PRINTS0,NASISLIB
ERASE PRINTS0
PRINT LIST.PRINTMON(0),,,EDIT,Y
ERASE SOURCE.PRINTS0
ERASE SOURCE.PRINTMON
```

5. LINKREST

```
 PROCDEF LINKREST
ERASE NASISLIB(0)(NASISX)
CDS NASYSLIB(0)(NASISX),NASISLIB(0)
ERASE NASISLIB(0)(RTS11X10)
CDS NASYSLIB(0)(RTS11X10),NASISLIB(0)
ERASE NASISLIB(0)(NASISPRO)
CDS NASYSLIB(0)(NASISPRO),NASISLIB(0)
ERASE NASISLIB(0)(LISRMLF)
CDS NASYSLIB(0)(LISRMLF),NASISLIB(0)
ERASE NASISLIB(0)(RDBPRNTR)
CDS NASYSLIB(0)(RDBPRNTR),NASISLIB(0)
ERASE NASISLIB(0)(RDBACCUM)
CDS NASYSLIB(0)(RDBACCUM),NASISLIB(0)
ERASE NASISLIB(0)(RUSERID)
CDS NASYSLIB(0)(RUSERID),NASISLIB(0)
ERASE NASISLIB(0)(STATIC)
CDS NASYSLIB(0)(STATIC),NASISLIB(0)
ERASE NASISLIB(0)(TRNSCT)
CDS NASYSLIB(0)(TRNSCT),NASISLIB(0)
ERASE NASISLIB(0)(TRNSCT#)
CDS NASYSLIB(0)(TRNSCT#),NASISLIB(0)
ERASE NASISLIB(0)(RDBVS)
CDS NASYSLIB(0)(RDBVS),NASISLIB(0)
ERASE NASISLIB(0)(RDBMLF)
```

```
CDS NASYSLIB(0)(RDBMLF),NASISLIB(0)
ERASE NASISLIB(0)(RDBDRIVE)
CDS NASYSLIB(0)(RDBDRIVE),NASISLIB(0)
```

6. LINKRET

```
 PROCDEF LINKRET
DEFAULT LINKWMSG=N
VV LINK.SOURCE.RETRIEVL,SOURCE.RETRIEVL
VV LINK.SOURCE.RETRIEV0,SOURCE.RETRIEV0
DDEF NASYSLIB,VP,NASYSLIB(0),,,,,,OLD
DDEF RETRIEV0,VP,RETRIEV0
ERASE RETRIEV0
DDEF RETRIEV0,VP,RETRIEV0,,,,,,NEW,,TLU
DISPLAY ' INCLUDING'
LNK RETRIEV0,Y,RETRIEV0,,N,N,N
REL NASYSLIB
DDEF NASISLIB,VP,NASISLIB(0),,,,,,OLD
ERASE NASISLIB(0)(RETRIEVL)
DISPLAY ' COMBINING'
LNK RETRIEVL,Y,NASISLIB,,N,Y,Y
REL RETRIEV0,NASISLIB
ERASE RETRIEV0
PRINT LIST.RETRIEVL(0),,,EDIT,Y
ERASE SOURCE.RETRIEVL
ERASE SOURCE.RETRIEV0
```

7. LINKSUPP

```
 PROCDEF LINKSUPP
DEFAULT LINKWMSG=N
VV LINK.SOURCE.SUPPORT,SOURCE.SUPPORT
VV LINK.SOURCE.SUPPORT0,SOURCE.SUPPORT0
DDEF NASYSLIB,VP,NASYSLIB(0),,,,,,OLD
DDEF SUPPORT0,VP,SUPPORT0
ERASE SUPPORT0
DDEF SUPPORT0,VP,SUPPORT0,,,,,,NEW,,TLU
DISPLAY ' INCLUDING'
LNK SUPPORT0,Y,SUPPORT0,,N,N,N
REL NASYSLIB
DDEF NASISLIB,VP,NASISLIB(0),,,,,,OLD
ERASE NASISLIB(0)(SUPPORT)
DISPLAY ' COMBINING'
LNK SUPPORT,Y,NASISLIB,,N,Y,Y
REL SUPPORT0,NASISLIB
ERASE SUPPORT0
PRINT LIST.SUPPORT(0),,,EDIT,Y
ERASE SOURCE.SUPPORT
ERASE SOURCE.SUPPORT0
```

8. NASISASM

```
 PROCDEF NASISASM
```

```
PARAM $01,$02
VV NASIS.SOURCE.$01,SOURCE.$01
DDEF ASMMAC,VI,ASMMAC(0),DISP=OLD
DDEF ASMNDX,VS,ASMNDX(0),DISP=OLD
DDEF RSOURCE,VI,RSOURCE,DISP=OLD
DDEF RINDEX,VS,RINDEX,DISP=OLD
DDEF ASMLIB,VP,MODLIB(0),OPTION=JOBLIB;-
ERASE MODLIB(0)($01)
DISPLAY '(ASSEMBLY INITIATED $01)'
ASM $01,Y,(ASMMAC,ASMNDX,RSOURCE,RINDEX),,-
N,N,Y,Y,N,N,Y,Y
PRINT LIST.$01(0),,,EDIT,Y
RELEASE ASMLIB
RELEASE ASMMAC
RELEASE ASMNDX
RELEASE RSOURCE
RELEASE RINDEX
DSS? SOURCE.$01
POD? MODLIB(0),Y,Y,$01
```

9. NASISASX

```
 PROCDEF NASISASX
PARAM $01,$02
VV NASIS.SOURCE.$01,SOURCE.$01
DDEF ASMMAC,VI,ASMMAC(0),DISP=OLD
DDEF ASMNDX,VS,ASMNDX(0),DISP=OLD
DDEF ASMLIB,VP,MODLIB(0),OPTION=JOBLIB;-
ERASE MODLIB(0)($01)
DISPLAY '(ASSEMBLY INITIATED $01)'
ASM $01,Y,(ASMMAC,ASMNDX),,N,N,Y,Y,N,N,Y,Y
PRINT LIST.$01(0),,,EDIT,Y
RELEASE ASMLIB
RELEASE ASMMAC
RELEASE ASMNDX
DSS? SOURCE.$01
POD? MODLIB(0),Y,Y,$01
```

10. NASISIN

```
 PROCDEF NASISIN
PARAM $01
ERASE SOURCE.$01
VV NASIS.SOURCE.$01,SOURCE.$01
```

11. NASISOUT

```
 PROCDEF NASISOUT
PARAM $01,$02
RELEASE BACKUPDD
IF '$02'  = '';ERASE NASIS.SOURCE.$01;-
DDEF BACKUPDD,VI,NASIS.SOURCE.$01,DISP=NEW,-
UNIT=(DA,2314),VOLUME=(,SAFETY);-
```

```
VV SOURCE.$01,NASIS.SOURCE.$01
IF '$02' ¬= '';ERASE NASIS.$01;-
DDEF BACKUPDD,$02,NASIS.$01,DISP=NEW,-
UNIT=(DA,2314),VOLUME=(,SAFETY)
IF '$02' ¬= '' & ( '$01' = 'MODLIB' | '$01' =
'NASYSLIB' );VV $01(0),NASIS.$01
IF '$02' ¬= '' & '$01' ¬= 'MODLIB' & '$01' ¬=
'NASYSLIB' ;VV $01,NASIS.$01
RELEASE BACKUPDD
```

12. NASISPLI

```
 PROCDEF NASISPLI
PARAM $01
VV NASIS.SOURCE.$01,SOURCE.$01
DDEF LISRMAC,VP,SOURCE.LISRMAC,DCB=-
(RECFM=V,KEYLEN=7,RKP=4,LRECL=255)
CATALOG GDG=LIST.$01,1,0,Y
DDEF PLIIST,VS,LIST.$01(+1),DCB=-
(RECFM=V,LRECL=133),DISP=NEW,RET=TLU
DDEF PLIILIB,VP,MODLIB(0),OPTION=JOBLIB
PLI $01,(NT,M,O=2,SM=(2,72,1),LC=60,-
A,X,L,E),(NOPRINT)
PRINT LIST.$01(0),,,EDIT,Y
RELEASE PLILIB
RELEASE LISRMAC
ERASE LOAD.$01
DELETE LOAD.$01
ERASE MAC.$01
DELETE MAC.$01
DEFAULT SYSINX=E
WHI  MODLIB(0)
PAT $01+4 00
RUN
DEFAULT SYSINX=G
DSS? SOURCE.$01
POD? MODLIB(0),Y,Y,$01
```

13. SEQUENCE

There exists (in UTILITYS) a program called SEQPGM. The function of the program is to date stamp lines of a modified program or sequence number new source decks. The program always places the current date in columns 73-80 of the first line of the dataset. Further, it makes every line appear as though it were generated from a card, i.e., post the keyboard/cardboard indicator, post the length to 92, pad with blanks through column 72 (or truncate) and post the sequence field. In the update mode the

current date is posted in the sequence field of any line that is not exactly 80 characters long or which does not indicate a card origin. In the sequence mode a unique program ID is placed into columns 73-76 of each line after the first, the dataset is rekeyed from line 100 in increments of 100, and digits 2-5 of the line number are posted into cclumns 77-80 of each line. The following command should be executed prior to using the program for the first time,

```
BUILTIN   SEQ, SEQPARMS.
```

The format for the program is,

```
SEQ  deckname, deckid
```

where:

deckname - is the fully qualified dsname of the dataset to be processed,

deckid - indicates by its presence that the program is to run in sequence mode and that this 1-4 character id is to be posted into columns 73-76 of all cards following the first.

EXAMPLE:

```
USER:  SEQ SOURCE.SEQPGM,SEQ
SYSTEM:  THE FOLLOWING LINES HAVE BEEN
TRUNCATED - 0012000
SYSTEM:  OK.
```

(at this point the user may examine dataset SEQ.ERROR.DATASET which will ccntain, intact, all lines indicated to have been truncated.)

E. Module Modification Procedure

This procedure for the modification and testing of the NASIS system modules is now in effect. The procedure covers the addition of new modules, as well as the modification of existing NASIS modules. The procedure can be logically divided into four phases - preparation, modification, testing, and insertion.

1. Preparation

The preparation phase begins with the release of the module for modification by the system maintenance task leader. Care must be taken to insure that no more than one person is concurrently modifying a particular module. Following this manual step, the source code for the module is copied, using the NASISIN procdef, from the master copy (NASIS.SOURCE.module) to the working copy (SOURCE.module) under the TSS userid SALISR.

At this point the implementor should use the following commands to gain access to the module under his TSS userid:

```
LOGON  userid
DELETE NASIS.SOURCE
SHARE  NASIS.SOURCE,SALISR,SOURCE
ERASE  SOURCE.module
VV     NASIS.SOURCE.module,-
       SOURCE.module
```

The source code for the module is now ready for modification by the implementor.

2. Modification

The modification phase will consist of the modification of the source code by the implementor followed by the re-compilation or re-assembly of the code. All modified or new lines of source code should be date stamped. This can be accomplished using the SEQ program previously described. Preliminary testing of the module will take place under the implementor's userid. The commands necessary for re-compilation are:

```
DDEF LISRMAC,VP,NASIS.SOURCE.LISRMAC,-
DCB=(RKP=4,RECFM=V,LRECL=255,-
KEYLEN=7)
DDEF JOBLIB,VP,joblib,OPTION=JOBLIB
PLI module,(O=02,SM=(2,72,1),LC=60,-
M,NT,A,X,L,E)
```

The commands necessary for re-assembly are:

```
DDEF ASMMAC,VI,ASMMAC(0),DISP=OLD
DDEF ASMNDX,VS,ASMNDX(0),DISP=OLD
DDEF RSOURCE,VI,RSOURCE,DISP=OLD
DDEF RINDEX,VS,RINDEX,DISP=OLD
DDEF JOBLIB,VP,joblib,OPTION=JOBLIB
```

```
ASM module,Y,(RSOURCE,RINDEX,ASMMAC,-
ASMNDX),,N,N,Y,Y,N,N,Y
```

It is assumed that the macro libraries have been properly shared before issuing the commands listed above.

3. Testing

To test the new module with the other components of the NASIS system, the implementor must gain access to the un-LINKEDITed version of the system. This can be accomplished by the following commands:

```
DELETE NASISLIB
SHARE NASISLIB,SALISR,NASYSLIB
```

After modifying or adding any BUILTINs, PROCDEFs or DEFAULTs required for the modified module the implementor is ready to test. The following DDEFs will cause the modules involved in the test to be loaded in the proper sequence.

```
DDEF NASISLIB,VP,NASISLIB(0),-
OPTION=JOBLIB
DDEF JOBLIB,VP,joblib,OPTION=JOBLIB
```

Any new or modified messages should be inserted into member LISRMLF of NASYSLIB(0) before testing.

4. Insertion

When the module has been completely tested and 'debugged' and is ready for insertion into the system, the implementor should inform the system maintenance task leader. He must then give TSS userid SALISR shared access to the modified source code and indicate to system maintenance any changes in the BUILTINs, PROCDEFs, DEFAULTs or messages used by the module.

System maintenance will then perform the following operations, under TSS userid SALISR, to insert the updated module in the system.

a. Compile or assemble the module with the appropriate options, placing the output

intc MODLIB(0), using the NASISASM, NASISASX, or NASISPLI procdefs.

b. After a clean compile or assembly copy the object module to NASYSLIB(0).

c. Copy the new object code to NASIS.NASYSLIB.

d. Update the master copy of the source code. Use the NASISOUT procdef.

e. Make any changes necessary to the text of USERJCIN and its backup, or to the message file and its backup, to reflect the new requirements of the module.

f. Inform the implementor that he must erase his copy of the module source code.

g. LINKEDIT the new or modified module into NASISLIB(0). This is discussed in Section II, Topic I of this document.

F. Joining Users

The USERJOIN procedure is to be used to join all users of the NASIS system. It includes all aspects of the system. The procedure for executing USERJOIN is as follows:

The first time that a TSS-ID is to be JOINed, type in the following commands:

1. RELEASE SYSULIB
2. CATALOG USERLIB,U,U,NEW.NASIS.SYSTEM.USERLIB
3. SHARE USERLIB,SALISR,USERJOIN
4. ABEND

The rest of the procedure is automatic. In addition, USERJCIN creates a procedure called REJOIN which contains the above commands. Therefore, subsequent joins can be effected by simply typing REJOIN.

NOTE: a second ABEND is prompted for, and MUST be entered to restore your original USERLIB.

The fcllowing is an example of this procedure:

>>>>>>>> LOGCN OR HANG UP.

```
LOGON SAISR3,,24,,X
 TSS/360 LEVEL 8.1
 ENTER PASSWORD
 XXXXXXX
  TASKID=006C  LOGON AT 10:20 ON 10/15
 ...RELEASE SYSULIB
 ...CATALOG USERLIB,U,U,NEW.NASIS.SYSTEM.USERLIB
 ...SHARE USERLIB,SALISR,USERJOIN
 ...ABEND

NASIS JOIN INITIATED.
   NASIS MESSAGES INSERTED
   NASIS PROCEDURES CREATED.
   NASIS FILES SHARED.
NASIS JOIN COMPLETED.
 BSN=0359
NOTE: A USERID ANALYSIS IS BEING EXECUTED
      SO THAT YOU MAY EXAMINE THE RESULTS
      OF THE JOIN PROCEDURE.
*** TYPE ABEND TO RESTORE YOUR USERLIB ***
TASK DELETED BY COMMAND
 NEW TASK LOGGED ON AT 10:32 ON 10/15/71. TASKID =
006F
_ABEND

TASK DELETED BY COMMAND
 NEW TASK LOGGED ON AT 10:33 ON 10/15/71. TASKID =
0071
 ...
```

G. Reserved Datasets, Modules, and Procdefs.

All NASIS system external names are reserved for those TSS-IDs joined to the system. The reserved procdef, builtin, synonym, default, libraries and dataset names can be determined by perusal of Appendices A, B, and C. The TSS-ID SALISR** has other restrictions on its datasets and procdefs which can be determined by perusal of Section C and Appendix D.

H. How to program new modules and new commands.

In order to program new modules and/or commands to work in conjunction with the NASIS system, certain minimum requirements must be met. These are as follows:

a. A working knowledge of TSS.
b. A working knowledge of PL/I.
c. A working knowledge of DBPL/I.
d. A working knowledge of TSPL/I

e. A working knowledge of NASIS.

1. Introduction

The design of the NASIS system is such that special retrieval commands may be designed and implemented in order to extend or refine the system for a limited user population. This document will identify the steps to be taken to effect the addition of the new command. It is assumed that the design and implementation of the command make use of the capabilities of DBPAC and TS in the proper fashion.

2. Implementation

After the new command modules have been designed and coded, the user should define a job library to house the new command modules. He should then share the NASIS PL/I macro library to obtain access to the preprocessors and any of the data tables that are required by the new command. The command may then be compiled.

EXAMPLE:

```
USER:  SHARE NASIS.SOURCE,SALISR,SOURCE
       (only once)
USER:  DDEF MYLIB,VP,MYLIB,OPTION=JOBLIB
USER:  DDEF LISRMAC,VP,NASIS.SOURCE.LISRMAC
USER:  PLI ACOMMAND, ...
       (supply PL/I options)
```

NOTE: The last three steps should be repeated as many times as necessary to obtain an error free compilation.

The next step is to define the command to the system. This can be accomplished by entering SYNONYM myverb=*COMMAND.

where:

myverb - identifies the command to be added.

Specified as: a 1-8 character name.

Next, assuming the command is a retrieval command, the user must modify the default symbol for user

specified retrieval commands, RETVERBS. This can be accomplished by entering

```
DEFAULT RETVERBS='myverb=mypgm'
```

where:

mypgm - identifies the entry point to be called when the command is encountered.

Finally, the changes should be stored in the user's permanent profile by entering PROFILE.

EXAMPLE:

The following example illustrates the above steps as they relate to the addition of the 'FIELDS' command to the retrieval system.

```
USER:  EDIT SOURCE.RDBFLDS
SYSTEM:  0000100  (user enters source
statements)
                .
                .
                .
USER:  _END
USER:  DDEF MODLIB,VP,MODLIB(0),OPTION=JOBLIB
USER:  DDEF LISRMAC,VP,SOURCE.LISRMAC
USER:  PLI RDBFLDS,(O=02,M,NT,SM=(2,72,1),-
IC=60,A,X,L,E)
SYSTEM:      (compiles the program)
USER:    BEGIN NASIS
SYSTEM:  -ENTER NASIS COMMAND:
USER:    SYNONYM FIELDS=*COMMAND;-
DEFAULT RETVERBS='FIELDS=DBFLDS';-
PROFILE.
```

NOTE: At execution time the job library must be defined before entering the 'BEGIN NASIS' command, so that the modules may be properly loaded.

If the modules require any prompting or diagnostic messages, these should be inserted into member LISRMLP of NASYSLIB(0).

I. LINKEDITing NASIS

LINKEDITing NASIS is done to accomplish two goals. First, to improve the running speed of the system.

Secondly, to reduce the size of the object library, and thus increase the loading time of the system.

1. When to LINKEDIT

   NASISLIB(0) is the LINKEDITed experimental version of NASIS. It is usually being tested for approval as the new running MTT system. Therefore, LINKEDITing into this library should only be done after consulting with your DBA.

2. How to LINKEDIT

   Procdefs have been created to make LINKEDITing as automatic as possible. These procdefs are all listed in Section II, Topic I. Basically these procdefs do the following:

   LINKALL - LINKEDITs all components of NASIS.

   LINKMAIN - LINKEDITS only the Maintenance Subsystem.

   LINKOUT - is used to backup all LINKEDIT source and object modules.

   LINKPRNT - LINKEDITs the Batch Print Monitor.

   LINKREST - copies all miscellaneous modules into NASYSLIB(0).

   LINKRET - LINKEDITs the Retrieval Subsystem.

   LINKSUPP - LINKEDITs all support modules.

To LINKEDIT NASIS you must:

   a. Insure that your task is in '32-bit' mode. Use the SET32 procdef.

   b. Enter the appropriate LINK.... procdef.

   c. Backup the results.

   EXAMPLE:

```
SET32
LINKALL
LINKOUT NASISLIB,VP
```

APPENDIX A.

THE USERID ANALYSIS

```
LOGON TSS
'
'
' *****.....STANDARD USERID ANALYSIS..(09/09/71).....***** '
'
'
$SPACE
USAGE
$SPACE
PROCDEFS
$SPACE
BUILTINS
$SPACE
SYNONYMS
$SPACE
DEFAULTS
$SPACE
POD? USERLIB
$SPACE
PC?
$SPACE
EDIT USERLIB(SYSMLF)
_LIST
END
$SPACE
EDIT USERLIB(SYSPRO)
_LIST
END
$SPACE
POD? USERLIB,Y,Y,*ALL
$SPACE
DSS?
$SPACE
$APPEND
PROCDEF TELLUSER
_EXCISE 100,LAST
_INSERT 100
PARAM $01,USERID=$02
PHONE $02,'*** $01 ***'
_END
TELLUSER 'USERID ANALYSIS COMPLETE'
PROCDEF TELLUSER
EXCISE 0,LAST
_END
LOGOFF
```

APPENDIX B.

THE USERID CREATION

```
' *****.....STANDARD USERID CREATION..(09/09/71).....***** '
'                                                           '
'                                                           '
' **********.....DEFINE MSG IN ZLOGON.....********** '
PROCDEF ZLOGON
_EXCISE 0,LAST
END
PROCDEF ZLOGON
ZZLOGON
DISPLAY 'USERID CREATION FAILED IN EXECUTION.'
DISPLAY 'RE-ENTER BACK COMMAND.'
_END
'                                                           '
DEFAULT LIMEN=N
DEFAULT LINENO=N
DEFAULT DEPROMPT=N
'                                                           '
'                                                           '
' *****.....DEFINE TEMPORARY PROCDEF ISRSHARE.....***** '
'                                                           '
PROCDEF ISRSHARE
PARAM DATASET,OWNERID
SET OD='OWNERID*******'
IF SYSTCM.(8,8)¬=OD;DELETE DATASET
IF SYSTCM.(8,8)¬=OD;ERASE DATASET
IF SYSTCM.(8,8)¬=OD;DELETE DATASET
IF SYSTCM.(8,8)¬=OD;SHARE DATASET,OWNERID,DATASET
_END
'                                                           '
'                                                           '
' *****.....EXCISE STANDARD USERID BUILTINS.....***** '
'                                                           '
PROCDEF BUILTINS
_EXCISE 0,LAST
PROCDEF CHECK
EXCISE  0,LAST
PROCDEF CONEDIT
_EXCISE 0,LAST
PROCDEF DEFAULTS
_EXCISE 0,LAST
PROCDEF DICT
_EXCISE 0,LAST
PROCDEF DS
_EXCISE 0,LAST
PROCDEF DM
_EXCISE 0,LAST
PROCDEF ITC
_EXCISE 0,LAST
```

```
PROCDEF LOOKIE
_EXCISE 0,LAST
PROCDEF OWNS
_EXCISE 0,LAST
PROCDEF PLI8
_EXCISE 0,LAST
PROCDEF PROCDEFS
_EXCISE 0,LAST
PROCDEF PRT
_EXCISE 0,LAST
PROCDEF PT
_EXCISE 0,LAST
PROCDEF REDIT
_EXCISE 0,LAST
PROCDEF SEQ
_EXCISE 0,LAST
PROCDEF SORT
_EXCISE 0,LAST
PROCDEF SYNONYMS
_EXCISE 0,LAST
PROCDEF USERS
_EXCISE 0,LAST
PROCDEF VI
_EXCISE 0,LAST
PROCDEF WHIZ
_EXCISE 0,LAST
'                                                                    '
'                                                                    '
'                                                                    '
' *****..........DELETE POSSIBLE SYNONYMS..........*****            '
SYNONYM BUILTINS=
SYNONYM CHECK   =
SYNONYM CONEDIT =
SYNONYM DEFAULTS=
SYNONYM DICT    =
SYNONYM DS      =
SYNONYM DM      =
SYNONYM ITC     =
SYNONYM LOOKIE  =
SYNONYM OWNS    =
SYNONYM PLI8    =
SYNONYM PROCDEFS=
SYNONYM PRT     =
SYNONYM PT      =
SYNONYM REDIT   =
SYNONYM SEQ     =
SYNONYM SORT    =
SYNONYM SYNONYMS=
SYNONYM USERS   =
SYNONYM VI      =
SYNONYM WHIZ    =
'                                                                    '
'                                                                    '
```

```
' *****.......DEFINE STANDARD USERID BUILTINS.........***** '
'                                                            '
BUILTIN CHECK    ,TIMER
BUILTIN CONEDIT ,CONEDIT
BUILTIN DICT     ,CZDICE
BUILTIN DEFAULTS,DEFAULTS
BUILTIN DS       ,DISPARMS
BUILTIN DM       ,DUMPARMS
BUILTIN ITC      ,CZOCTAI1
BUILTIN LOOKIE   ,ISLAB
BUILTIN OWNS     ,OWNEP
BUILTIN PLI8     ,CFEAA8
BUILTIN PROCDEFS,PROCDEFS
BUILTIN PRT      ,PRTPARMS
BUILTIN PT       ,PATPARMS
BUILTIN REDIT    ,REDIT
BUILTIN SEQ      ,SEQPARMS
BUILTIN SORT     ,SRTBPKD
BUILTIN SYNONYMS,SYNCNYMS
BUILTIN USERS    ,CZAID1E
BUILTIN VI       ,VIPARAMS
BUILTIN WHIZ     ,WHIZPARM
BUILTIN BUILTINS,BUILTINS
'                                                            '
'                                                            '
' *****.......EXCISE STANDARD USERID SYNONYMS.........***** '
'                                                            '
PROCDEF B
_EXCISE 0,LAST
PROCDEF CAN
_EXCISE 0,LAST
PROCDEF CAT
_EXCISE 0,LAST
PROCDEF CON
_EXCISE 0,LAST
PROCDEF COR
_EXCISE 0,LAST
PROCDEF D
_EXCISE 0,LAST
PROCDEF DD
_EXCISE 0,LAST
PROCDEF DDS
_EXCISE 0,LAST
PROCDEF DEF
_EXCISE 0,LAST
PROCDEF DIE
_EXCISE 0,LAST
PROCDEF DSS
_EXCISE 0,LAST
PROCDEF ER
_EXCISE 0,LAST
PROCDEF EX
_EXCISE 0,LAST
```

```
PROCDEF EXP
_EXCISE 0,LAST
PROCDEF GET
_EXCISE 0,LAST
PROCDEF I
_EXCISE 0,LAST
PROCDEF JBS
_EXCISE 0,LAST
PROCDEF L?
_EXCISE 0,LAST
PROCDEF LO
_EXCISE 0,LAST
PROCDEF MOD
_EXCISE 0,LAST
PROCDEF NUM
_EXCISE 0,LAST
PROCDEF OFF
_EXCISE 0,LAST
PROCDEF PC
_EXCISE 0,LAST
PROCDEF PER
_EXCISE 0,LAST
PROCDEF POD
_EXCISE 0,LAST
PROCDEF PR
_EXCISE 0,LAST
PROCDEF PROC
_EXCISE 0,LAST
PROCDEF PROF
_EXCISE 0,LAST
PROCDEF PU
_EXCISE 0,LAST
PROCDEF Q
_EXCISE 0,LAST
PROCDEF REG
_EXCISE 0,LAST
PROCDEF REL
_EXCISE 0,LAST
PROCDEF REM
_EXCISE 0,LAST
PROCDEF REV
_EXCISE 0,LAST
PROCDEF SEC
_EXCISE 0,LAST
PROCDEF SH
_EXCISE 0,LAST
PROCDEF SYN
_EXCISE 0,LAST
PROCDEF UNL
_EXCISE 0,LAST
PROCDEF UPD
_EXCISE 0,LAST
PROCDEF X
```

```
_EXCISE 0,LAST
PROCDEF XBT
_EXCISE 0,LAST
'                                                                    '
'                                                                    '
' *****.......DEFINE STANDARD USERID SYNONYMS........***** '
'                                                                    '
SYNONYM B        =BRANCH   ,CAT      =CATALOG
SYNONYM CON      =CONTEXT  ,COR      =CORRECT  ,D        =DISPLAY
SYNONYM DD       =DDEF     ,DDS      =DDNAME?  ,DEF      =DEFAULT
SYNONYM DIE      =ABEND    ,DSS      =DSS?
SYNONYM EX       =EXECUTE  ,EXP      =EXPLAIN  ,GET      =EXCERPT
SYNONYM I        =INSERT   ,JBS      =JOBLIBS  ,L?       =LINE?
SYNONYM LO       =LOCATE   ,MOD      =MODIFY   ,NUM      =NUMBER
SYNONYM OFF      =LOGOFF   ,PC       =PC?      ,PER      =PERMIT
SYNONYM POD      =POD?     ,PR       =PRINT    ,PROC     =PROCDEF
SYNONYM PROF     =PROFILE  ,PU       =PUNCH    ,Q        =QUALIFY
SYNONYM REG      =REGION   ,REM      =REMOVE
SYNONYM REV      =REVISE   ,SEC      =SECURE   ,SH       =SHARE
SYNONYM SYN      =SYNONYM  ,UNL      =UNLOAD   ,UPD      =UPDATE
SYNONYM X        =EXCISE   ,XBT      =EXHIBIT
'                                                                    '
'                                                                    '
' *****.......DEFINE STANDARD USERID DEFAULTS........***** '
'                                                                    '
DEFAULT ACCESS   =RO
DEFAULT BREVITY =T
DEFAULT BRIEF    =MP
DEFAULT LINES    =60
DEFAULT PMDLIST =Y
DEFAULT STORED   =Y
DEFAULT PLIOPT   =(O=02,M,NT,SM=(2,72,1),LC=60,A,X,L,E)
DEFAULT PLIPACK =N
DEFAULT PLCOPT   =(NOPRINT)
DEFAULT BATCHOUT='**********.............BATCH OUTPUT -
SEPARATOR.............**********'
'                                                                    '
'                                                                    '
' *****.......EXCISE STANDARD USERID PROCDEFS........***** '
'                                                                    '
PROCDEF $SPACE
_EXCISE 0,LAST
PROCDEF BATCH
_EXCISE 0,LAST
PROCDEF BATCH$$1
_EXCISE 0,LAST
PROCDEF BATCH$$2
_EXCISE 0,LAST
PROCDEF BATCH$$3
_EXCISE 0,LAST
PROCDEF BATCH$$4
_EXCISE 0,LAST
PROCDEF BATCH$$5
```

```
_EXCISE 0,LAST
PROCDEF CAN
_EXCISE 0,LAST
PROCDEF DA
_EXCISE 0,LAST
PROCDEF DEL
_EXCISE 0,LAST
PROCDEF DIS
_EXCISE 0,LAST
PROCDEF DR
_EXCISE 0,LAST
PROCDEF DUM
_EXCISE 0,LAST
PROCDEF ER
_EXCISE 0,LAST
PROCDEF HALT
_EXCISE 0,LAST
PROCDEF ID
_EXCISE 0,LAST
PROCDEF LPROC
_EXCISE 0,LAST
PROCDEF PAT
_EXCISE 0,LAST
PROCDEF PLIT
_EXCISE 0,LAST
PROCDEF REL
_EXCISE 0,LAST
PROCDEF RENAME
_EXCISE 0,LAST
PROCDEF RETREGS
_EXCISE 0,LAST
PROCDEF SETREGS
_EXCISE 0,LAST
PROCDEF SET24
_EXCISE 0,LAST
PROCDEF SET32
_EXCISE 0,LAST
PROCDEF ZZLOGCN
_EXCISE 0,LAST
;                                                                      ;
;                                                                      ;
;                                                                      ;
; *****..........DELETE POSSIBLE SYNONYMS.....*****                   ;
;                                                                      ;
SYNONYM $SPACE  =
SYNONYM BATCH   =
SYNONYM BATCH$$1=
SYNONYM BATCH$$2=
SYNONYM BATCH$$3=
SYNONYM BATCH$$4=
SYNONYM BATCH$$5=
SYNONYM CAN     =
SYNONYM DA      =
```

```
SYNONYM DEL      =
SYNONYM DIS      =
SYNONYM DR       =
SYNONYM DUM      =
SYNONYM ER       =
SYNONYM HALT     =
SYNONYM ID       =
SYNONYM IPROC    =
SYNONYM PAT      =
SYNONYM PLIT     =
SYNONYM REL      =
SYNONYM RENAME   =
SYNONYM RETREGS  =
SYNONYM SETREGS  =
SYNONYM SET24    =
SYNONYM SET32    =
SYNONYM ZZLOGON  =
'                                                                    '
'                                                                    '
' *****.......DEFINE STANDARD USERID PROCDEFS........***** '
'                                                                    '
PROCDEF $APPEND
_END
'                                                                    '
PROCDEF $SPACE
PARAM BATCHOUT=$01
DISPLAY '      $C1'
DISPLAY '      $01'
DISPLAY '      $C1'
DISPLAY '      $01'
DISPLAY '      $C1'
_END
'                                                                    '
PROCDEF BATCH
PARAM $01,$02,$03
IF '$01'='';DISPLAY 'CANCELLED: PROCEDURE NAME MISSING.'
IF '$01'¬='' & ('$02'='I' | '$02'='E');BATCH$$4 $01,$02
IF '$01'¬='';DEFAULT BATCHPRO=$01,BATCHPRM=
IF ('$01'¬='' & '$03'¬='');DEFAULT BATCHPRM=',$03'
IF '$01'¬='';BATCH$$1
_END
'                                                                    '
PROCDEF BATCH$$1
PARAM BATCHPRO=$01,LIMEN=$W,LINENO=$Y
DEFAULT LINENO=N,LIMEN=N,SYSINX=E
PROCDEF BATCH$$2
__REVISE 200
__EXCERPT BATCH.$01.NAMES,,100
__END
DEFAULT LIMEN=$W,LINENO=$Y,SYSINX=G
SET BATCHBIT=0
BATCH$$2
IF BATCHBIT=0;ERASE BATCH.$01.NAMES;SET BATCHBIT=1
```

```
_END
'                                                                    '
PROCDEF BATCH$$2
DEFAULT BATCH$ID=-
BATCH$$2
BATCH$$3
_END
'                                                                    '
PROCDEF BATCH$$3
PARAM BATCH$ID=$01,BATCHPRC=$02,BATCHPRM=$03
IF ¬('$01'='**END**' | '$01'='**SKIP**');$02 $01$03
IF '$01'¬='**END**';BATCH$$5
IF '$01'¬='**END**';BATCH$$1
_END
'                                                                    '
PROCDEF BATCH$$4
PARAM $01,$02,LIMEN=$W,LINENO=$Y
DEFAULT LIMEN=N,LINENO=N,SYSINX=E
IF '$02'='I';ERASE BATCH.$01.NAMES
EDIT BATCH.$01.NAMES
__IF '$02'='E';CONTEXT 100,LAST,'**END**','**SKIP**'
__END
DEFAULT SYSINX=G
DISPLAY 'ENTER THE NAMES TO BE PROCESSED.'
EDIT BATCH.$01.NAMES
__INSERT LAST
DEFAULT SYSINX=E
EDIT BATCH.$01.NAMES
**END**
__END
DEFAULT SYSINX=G,LINENO=$Y,LIMEN=$W
_END
'                                                                    '
PROCDEF BATCH$$5
PARAM BATCHPRO=$01,LIMEN=$W,LINENO=$Y
DEFAULT LIMEN=N,LINENO=N,SYSINX=E
EDIT BATCH.$01.NAMES
__EXCISE 100
__NUMBER 200,LAST,100,100
__END
DEFAULT LIMEN=$W,LINENO=$Y,SYSINX=G
_END
'                                                                    '
PROCDEF CAN
PARAM
$01,$02,$03,$04,$05,$06,$07,$08,$09,$10,$11,$12,$13,$14,$15
CANCEL $01;IF '$02'¬='';-
CAN $02,$03,$04,$05,$06,$07,$08,$09,$10,$11,$12,$13,$14,$15
_END
'                                                                    '
PROCDEF DA
PARAM $01
DISPLAY A'$01'
```

```
_END
;
PROCDEF DEL
PARAM
$01,$02,$03,$04,$05,$06,$07,$08,$09,$10,$11,$12,$13,$14,$15
DELETE $01;IF '$02'¬='';-
DEL $02,$03,$04,$05,$06,$07,$08,$09,$10,$11,$12,$13,$14,$15
_END
;
PROCDEF DIS
PARAM $01,$02,$03
SETREGS
DS $01,$02,$03
_END
;
PROCDEF DR
PARAM $01,$02
IF '$01'='' & '$02'='';DISPLAY 0:15R
IF '$01'¬='' & '$02'¬='';DISPLAY $01:$02R
IF '$01'¬='' & '$02'='';DISPLAY $01R
IF '$01'='' & '$02'¬='';DISPLAY 0:$02R
_END
;
PROCDEF DUM
PARAM $01,$02,$03
SETREGS
DM $01,$02,$03
_END
;
PROCDEF ER
PARAM
$01,$02,$03,$04,$05,$06,$07,$08,$09,$10,$11,$12,$13,$14,$15
ERASE $01;IF '$02'¬='';-
ER $02,$03,$04,$05,$06,$07,$08,$09,$10,$11,$12,$13,$14,$15
_END
;
PROCDEF HALT
PARAM HALTADDR=$01,$02,$03
SET SAVE$$$1=0D,SAVE$$$2=2D,0D='$01          ',2D='0
'
IF '$01'¬='' & 0D > 2D;DEFAULT HALTADDR='$01';AT $01.-
(X'0$02');$03;STOP
IF 0D ¬> 2D;DEFAULT HALTADDR='$01';AT L'$01'.(X'0$02')-
$03;STOP
SET 0D=SAVE$$$1,2D=SAVE$$$2
_END
;
PROCDEF ID
PARAM $01
DISPLAY ID? L'$01'
_END
;
PROCDEF LPROC
```

```
PARAM $01,PRLINE=$02,SYSINX=$G,LIMEN=$W,LINENO=$Y
DEFAULT SYSINX=E,LIMEN=T
IF '$02'¬='';DEFAULT LINENC=N
DISPLAY '..... $01 .....'
EDIT USERLIB(SYSPRO)
__REGION $01
__LIST 0,LAST
__END
DEFAULT SYSINX=$G,LIMEN=$W,LINENO=$Y
_END
'                                                                    '
PROCDEF PAT
PARAM $01,$02,$03
SETREGS
PT $01,$02,$03
_END
'                                                                    '
PROCDEF PLIT
PARAM NAME=$01,PLIOPT=$02,PLCOPT=$03,SOURCEDS=$04,-
MACRODS=$05,TOPLIB=$06
DDEF PLILIB,VP,PLILIB,,,,,,OLD,JOBLIB
LOAD CFBAA
CATALOG GDG=LIST.$01,1,0,Y
DDEF PLILIST,VS,LIST.$01(+1),,,,,,NEW,RET=TLU
IF '$06'='';JOBLIBS SYSULIB
IF '$06'¬='';JOBLIBS $06
PLI8 NAME=$01,PLIOPT=$02,PLCOPT=$03,SOURCEDS=$04,MACRODS=$05
RELEASE PLILIB
_END
'                                                                    '
PROCDEF REL
PARAM
$01,$02,$03,$04,$05,$06,$07,$08,$09,$10,$11,$12,$13,$14,$15
RELEASE $01;IF '$02'¬='';-
REL $02,$03,$04,$05,$06,$07,$08,$09,$10,$11,$12,$13,$14,$15
_END
'                                                                    '
PROCDEF RENAME
PARAM $01,$02
CATALOG $01,U,U,$02
_END
'                                                                    '
 PROCDEF RETREGS
PARAM LIMEN=$01
DEFAULT LIMEN=W;SET
L'6E4':L'6E7'=REGS.(52,4),L'6EC':L'6F3'=REGS.(56,8)
SET L'6F4':L'727'=REGS.(0,52);DEFAULT LIMEN=$01
_END
'                                                                    '
PROCDEF SETREGS
PARAM LIMEN=$01
DEFAULT LIMEN=W;-
SET REGS.(52,4)=I'6E4:I'6E7',REGS.(56,8)=L'6EC':L'6F3'
```

```
SET REGS.(0,52)=L'6F4':I'727';DEFAULT LIMEN=$01
_END
'                                                                    '
PROCDEF SET24
SET L'692'=X'08'
ABEND
_END
'                                                                    '
PROCDEF SET32
SET L'692'=X'0A'
ABEND
_END
'                                                                    '
PROCDEF ZZLOGON
DDEF UTILITYS,VP,UTIIITYS(0),CPTICN=JOBLIB
JOBLIBS SYSULIB
_END
'                                                                    '
'                                                                    '
' *****...........SHARE UTILITYS DATASET.............*****         '
'                                                                    '
ISRSHARE UTILITYS,SAUTY
ISRSHARE PLILIB,SALISR
'                                                                    '
'                                                                    '
' *****......DELETE TEMPORARY PROCDEF ISRSHARE.......*****         '
'                                                                    '
PROCDEF ISRSHARE
_EXCISE 0,LAST
END
'                                                                    '
'                                                                    '
' *****...........FINISH JOB AND LOGOFF..............*****         '
'                                                                    '
DEFAULT LIMEN=W
DEFAULT LINENO=Y
DEFAULT DEPROMPT=Y
PROFILE
'                                                                    '
PROCDEF ZLOGON
_EXCISE 200,300
_END
PHONE SALISR,'USERID CREATICN COMPLETE.'
'                                                                    '
LOGOFF
```

APPENDIX C.

USERJOIN

..LISTING OF THE COMPONENTS OF THE NASIS SYSTEM USERJOIN..

********.....USERJOIN ZLOGON PROCEDURE.....********

```
PROCDEF ZLOGON
DISPLAY 'NASIS JOIN INITIATED.'
JOIN
RECORD
DISPLAY '   NASIS FILES SHARED.'
RELEASE SYSULIB
DELETE USERLIB
CATALOG NEW.NASIS.SYSTEM.USERLIB,U,U,USERLIB
DISPLAY 'NASIS JOIN COMPLETED.'
DELETE THE.USERID
SHARE THE.USERID,SALISR,THE.USERID
EXECUTE THE.USERID.ANALYSIS
DISPLAY 'NOTE: A USERID ANALYSIS IS BEING EXECUTED'
DISPLAY '      SO THAT YOU MAY EXAMINE THE RESULTS'
DISPLAY '      OF THE JOIN PROCEDURE.'
DISPLAY '*** TYPE ABEND TO RESTORE YOUR USERLIB ***'
```

*******.....USERJOIN JOIN PROCEDURE.....********

```
PROCDEF JOIN
PARAM LINEN=$W
DEFAULT SYSINX=E
ERASE NEW.NASIS.SYSTEM.USERLIB(SYSPRD)
EDIT NEW.NASIS.SYSTEM.USERLIB(SYSMLF)
_REGION CZCDL003;_REVISE 0
ISA $1 IN $2 UNDEFINED.
_REGION CZCDL005;_REVISE 0
ISA MODULE $1 HAS LEVEL $2 ERRORS.
_REGION CZCDL020;_REVISE 0
ISA PROCEEDING: ENTRY POINT $01, MODULE $02 DUPLICATES ENTRY
POINT IN MODULE $03.
_REGION CZCDL021;_REVISE 0
ISA PROCEEDING: CSECT $01, MODULE $02 DUPLICATES A CSECT IN
MODULE $03.
_REGION MTTSUP01;_EXCISE 0,LAST
_REGION MTTSUP02;_EXCISE 0,LAST
_REGION MTTSUP03;_EXCISE 0,LAST
_REGION MTTSUP04;_EXCISE 0,LAST
_REGION MTTSUP05;_EXCISE 0,LAST
_REGION MTTSUP06;_EXCISE 0,LAST
```

```
_REGION MTTSUP07;_EXCISE 0,LAST
_REGION MTTSUP08;_EXCISE 0,LAST
_REGION MTTSUP09;_EXCISE 0,LAST
_REGION MTTSUP10;_EXCISE 0,LAST
_REGION MTTSUP11;_EXCISE 0,LAST
_REGION MTTSUP12;_EXCISE 0,LAST
_REGION MTTSUP13;_EXCISE 0,LAST
_REGION MTTSUP14;_EXCISE 0,LAST
_REGION MTTSUP15;_EXCISE 0,LAST
_REGION MTTSUP16;_EXCISE 0,LAST
_REGION MTTSUP17;_EXCISE 0,LAST
_REGION MTTSUP18;_EXCISE 0,LAST
_REGION MTTSUP19;_EXCISE 0,LAST
_REGION MTTSUP20;_EXCISE 0,LAST
_REGION MTTSUP21;_EXCISE 0,LAST
_REGION MTTSUP22;_EXCISE 0,LAST
_REGION MTTSUP23;_EXCISE 0,LAST
_REGION MTTSUP24;_EXCISE 0,LAST
_REGION MTTSUP25;_EXCISE 0,LAST
_REGION MTTSUP99;_EXCISE 0,LAST
_END
DISPLAY '   NASIS MESSAGES INSERTED.'
EDIT NEW.NASIS.SYSTEM.USERLIB(SYSPRO)
_REGION BACKEXSE;_EXCISE 0,LAST
_REGION BACKSRCH;_EXCISE 0,LAST
_REGION $BEGIN;_EXCISE 0,LAST
_REGION ACCUM;_EXCISE 0,LAST;_INSERT 0,100
PROCDEF ACCUM
PARAM $01,$02,$03,$04,$05,$06,$07,$08,$09,$10
IF NASISMTT¬=1;DDEF
NASISLIB,VP,NASISLIB(0),DISP=CLD,OPTION=JOBLIB
CALL RDBACCUM,'$01,$02,$03,$04,$05,$06,$07,$08,$09,$10'
_REGION BEGIN;_EXCISE 0,LAST;_INSERT 0,100
PROCDEF BEGIN
PARAM $01
IF NASISMTT¬=1; DDEF NASISLIB,VP,NASISLIB(0),OPTION=JOBLIB
IF NASISMTT¬=1; DDEF DBALIB,VP,DBALIB(0),OPTION=JOBLIB
IF NASISMTT¬=1; LOAD NASISX; SET NASISMTT=1
IF '$01'='NASIS'; CALL NASIS1
IF '$01'='PRINTS'; CALL NASISP
_REGION CANSRCH;_EXCISE 0,LAST
_REGION COPY;_EXCISE 0,LAST;_INSERT 0,100
 PROCDEF COPY
PARAM OPTION=$01,NAME=$02
IF '$01'='PROFILE'; COPYPRO
IF '$01'='FORMAT' & '$02'¬='';  COPYFMT '$02'
IF '$01'='STRATEGY' & '$02'¬='';  COPYSTR '$02'
IF ¬('$01'='PROFILE' | '$01'='FORMAT' | '$01'='STRATEGY') |
('$01'='FORMAT' & '$02'='')-
| ('$01'='STRATEGY' & '$02'=''); DISPLAY 'CANCELLED: INVALID
OR MISSING PARAMETERS.'
_REGION COPYFMT;_EXCISE 0,LAST;INSERT 0,100
PROCDEF COPYFMT
```

```
PARAM REGION=$01,OWNERID=$02,OLDID=$03,NEWID=$04
DELETE ORIGINAL.STRATEGY.LIBRARY
SHARE ORIGINAL.STRATEGY.LIBRARY,$02,$03.STRATEGY.LIBRARY
CDS ORIGINAL.STRATEGY.LIBRARY( $01),$04.STRATEGY.LIBRARY
DELETE ORIGINAL.STRATEGY.LIBRARY
_REGION COPYPRO;_EXCISE 0,LAST;_INSERT 0,100
 PROCDEF COPYPRC
PARAM OWNERID=$01,OLDID=$02,NEWID=$03,LIMEN=$W
DEFAULT LIMEN=T
DELETE ORIGINAL.PROFILE.LIBRARY
SHARE ORIGINAL.PROFILE.LIBRARY,$01,NASIS.PROFILE.LIBRARY
ERASE INTERIM.PROFILE.LIBRARY
RELEASE PROCOPY1
DDEF PROCOPY1,VP,INTERIM.PROFILE.LIBRARY
CDS ORIGINAL.PROFILE.LIBRARY($02),INTERIM.PROFILE.LIBRARY
IF '$02'¬='$03'; WHIZPRC '$02','$03'
RELEASE PROCOPY2
DDEF PROCOPY2,VP,NASIS.PROFILE.LIBRARY
CDS
INTERIM.PROFILE.LIBRARY($03),NASIS.PROFILE.LIBRARY,REPLACE=R
ERASE INTERIM.PROFILE.LIBRARY
RELEASE PROCOPY2
DELETE ORIGINAL.PROFILE.LIBRARY
DEFAULT LIMEN=$W
_REGION COPYSTR;_EXCISE 0,LAST;INSERT 0,100
 PROCDEF COPYSTR
PARAM REGION=$01,OWNERID=$02,OLDID=$03,NEWID=$04
DELETE ORIGINAL.STRATEGY.LIBRARY
SHARE ORIGINAL.STRATEGY.LIBRARY,$02,$03.STRATEGY.LIBRARY
CDS ORIGINAL.STRATEGY.LIBRARY($01),$04.STRATEGY.LIBRARY
DELETE ORIGINAL.STRATEGY.LIBRARY
_REGION $CORRECT;_EXCISE 0,LAST
_REGION DELSTRAT;_EXCISE 0,LAST
_REGION ENTER;_EXCISE 0,LAST
_REGION EXPAND;_EXCISE 0,LAST
_REGION EXSEARCH;_EXCISE 0,LAST
_REGION FIELDS;_EXCISE 0,LAST
_REGION FINISH;_EXCISE 0,LAST
_REGION FORMAT;_EXCISE 0,LAST
_REGION GENERATE;_EXCISE 0,LAST
_REGION GFIELDS;_EXCISE 0,LAST
_REGION HEADER;_EXCISE 0,LAST
_REGION KEEP;_EXCISE 0,LAST
_REGION LATLON;_EXCISE 0,LAST
_REGION LIMIT;_EXCISE 0,LAST
_REGION MTTRESET;_EXCISE 0,LAST
_REGION NAME;_EXCISE 0,LAST
_REGION NASISMTT;_EXCISE 0,LAST;_INSERT 0,100
PROCDEF NASISMTT
PARAM $01
IF NASISMTT¬=1; DDEF NASISLIB,VP,NASISLIB(0),OPTION=JOBLIB
IF NASISMTT¬=1; DDEF DEALIB,VP,DEALIB(0),OPTION=JOBLIB
IF NASISMTT¬=1; DDEF MTTLIB,VP,MTTLIB(0),,,,,,OLD,JOBLIB
```

```
IF NASISMTT¬=1; LOAD NASIS; SET NASISMTT=1
IF '$01'='';  MTT NASIS,50,102,256
_REGION PAGE;_EXCISE 0,LAST
_REGION PRTDESC;_EXCISE 0,LAST
_REGION RECORD;_EXCISE 0,LAST
_REGION PRCOPIES;_EXCISE 0,LAST;INSERT 0,100
PROCDEF PRCOPIES
IF CCOUNT>0;PRINT LIST.STATS(0),PRTSP=EDIT;SET
CCOUNT=CCOUNT-1;PRCOPIES
_REGION REJOIN;_EXCISE 0,LAST;_INSERT 0,100
PROCDEF REJOIN
RELEASE SYSULIB
CATALOG USERLIB,U,U,NEW.NASIS.SYSTEM.USERLIB
SHARE USERLIB,SALISR,USERJOIN
ABEND
_REGION REPORT;_EXCISE 0,LAST;INSERT 0,100
PROCDEF REPORT
PARAM COPIES=$01
SET CCOUNT=1
IF '$01'¬='';SET CCOUNT=$01
IF NASISMTT¬=1;DDEF
NASISLIB,VP,NASISLIB(0),DISP=CLD,OPTION=JOBLIB
CATALOG GDG=LIST.STATS,1,0,Y
DDEF REPORT,VS,LIST.STATS(+1),DISP=NEW,RET=TLU
CALL FDBPRNTR
RELEASE REPORT
PRCOPIES
_REGION RERUN;_EXCISE 0,LAST
_REGION RESTART;_EXCISE 0,LAST
_REGION REVIEW;_EXCISE 0,LAST;_INSERT 0,100
PROCDEF REVIEW
PARAM $01,SYSINX=$G,LINENO=$Y
DEFAULT SYSINX=F,LINENO=N
EDIT NASIS.NEWS
__IF '$01'¬='';_REGION $01;_LIST 100,LAST
__IF '$01'='';_LIST
__END
DEFAULT SYSINX=$G,LINENO=$Y
_REGION SAVE;_EXCISE 0,LAST
_REGION SEARCH;_EXCISE 0,LAST
_REGION SELECT;_EXCISE 0,LAST
_REGION SHARING;_EXCISE 0,LAST;_INSERT 0,100
 PROCDEF SHARING
PARAM MODE=$01,NASISID=$02,LIMEN=$W
DEFAULT LIMEN=T
IF ¬('$01'='ON' | '$01'='OFF') | '$02'='';  DISPLAY
'CANCELLED: INVALID OR MISSING PARAMETERS.'
IF '$01'='ON' & '$02'¬='';  PERMIT
$02.STRATEGY.LIBRARY,*ALL,RO; -
PERMIT NASIS.PRCFILE.LIBRARY,*ALL,RO
IF '$01'='OFF' & '$02'¬='';  PERMIT
$02.STRATEGY.LIBRARY,*ALL,R; -
PERMIT NASIS.PRCFILE.LIBRARY,*ALL,R
```

```
DEFAULT LIMEN=$W
_REGION STORE;_EXCISE 0,LAST
_REGION STRATEGY;_EXCISE 0,LAST
_REGION TITLE;_EXCISE 0,LAST
_REGION USING;_EXCISE 0,LAST;_INSERT 0,100
PROCDEF USING
PARAM OWNERID=$01,OLDID=$02,NEWID=$03
IF ('$01'='' | '$02'=''); DISPLAY 'CANCELLED: INVALID OR
MISSING PARAMETERS.'
IF ¬('$01'='' | '$02'=''); DEFAULT OWNERID=$01; DEFAULT
OLDID=$02; DEFAULT NEWID=$03
IF ¬('$01'='' | '$02'='') & '$03'=''; DEFAULT NEWID=$02
_REGION WHIZPRO;_EXCISE 0,LAST;_INSERT 0,100
PROCDEF WHIZPRO
PARAM $01,$02,SYSINX=$G
DEFAULT SYSINX=E
WHIZ INTERIM.PROFILE.LIBRARY
STOW C $01,$02
RUN
DEFAULT SYSINX=$G
_REGION ZLOGON;_EXCISE 0,LAST;_INSERT 0,100
PROCDEF ZLOGON
SET NASISMTT=0
ZZLOGON
_END
DISPLAY '   NASIS PROCEDURES CREATED.'
ISRSHARE NASISLIB,SALISR,NASISLIB
DDEF NASISLIB,VP,NASISLIB(0),,,,,,OLD,JOBLIB
CALL RUSERID
ISRSHARE MTTLIB,SALISR
ISRSHARE NASIS.USERIDS,SADBA
ISRSHARE DBALIB,SADBA
ISRSHARE SAOWNER.DB2TDB,SADBA,SADBA.DB2TDB
ISREUILD TRNSCT
DEFAULT LIMEN=T
DELETE NASIS.STATIC
ERASE NASIS.STATIC
DEFAULT LIMEN=$W
DDEF NS,VI,NASIS.STATIC,DISP=NEW
CDS NASISLIB(0)(STATIC),NASIS.STATIC
RELEASE NS
ISRSHARE JOINIDS,SADBA
MODIDS
DELETE JOINIDS
DEFAULT SYSINX=G


*********.........USERJOIN RECORD PROCEDURE........*********


PROCDEF RECORD
PARAM USERID=$01,SYSINX=$E
DEFAULT SYSINX=E
```

```
EDIT NEW.NASIS.SYSTEM.USERLIB(SYSMLF)
_REGION NASISVID;_EXCISE 0,LAST;_INSERT 0
ISA $01 HAS BEEN JOINED TO NASIS VERSION 2.0 (12-01-72).
_END
DEFAULT SYSINX=$E
```

APPENDIX D.

NASIS MODULES

MODULE NAME	LANGUAGE
ANALYSIS	TSS
CREATION	TSS
LISRMAC	PLI
LISRMLF	ENG
MODLIB	NON
MTTLIB	N/A
NASISLIB	N/A
NASYSLIB	N/A
NASISPRO	N/A
NASISX	ASM
RCCLIST	PLI
RDBLIST	PLI
RDBACCUM	PLI
RDBATTN	PLI
RDBCLMN	PLI
RDBCMND	PLI
RDBCOMNT	PLI
RDBCORR	PLI
RDBDRIVE	PLI
RDBDSPL	PLI
RDBEDAC	PLI
RDBEDAR	PLI
RDBEDCM	PLI
RDBEDCP	PLI
RDBEDCS	PLI
RDBEDDE	PLI
RDBEDDI	PLI
RDBEDDL	PLI
RDBEDDP	PLI
RDBEDFD	PLI
RDBEDFI	PLI
RDBEDFS	PLI
RDBEDIN	PLI
RDBEDID	PLI
RDBEDMO	PLI
RDBEDPA	PLI
RDBEDPR	PLI
RDBEDRS	PLI
RDBEDRT	PLI
RDBEDRV	PLI
RDBEDSS	PLI
RDBEDSU	PLI
RDBEXITS	PLI
RDBEXPL	PLI
RDBEXSR	PLI
RDBFLDS	PLI

RDBFORM	PLI
RDBGENR	PLI
RDBGFLDS	PLI
RDBINDM2	PLI
RDBINIT	PLI
RDBJOIN	PLI
RDBLATI	PLI
RDBLDBK	PLI
RDBLIST	PLI
RDBLOAD	PLI
RDBMAIN	PLI
RDBMERGE	PLI
RDBMLF	PLI
RDBMNTN	PLI
RDBMTT	PLI
RDBPAC	PLI
RDBPLINK	ASM
RDBPRNT	PLI
RDBPRNTM	PLI
RDBPRNTR	PLI
RDBPRINT	PLI
RDBPRC	PLI
RDBSETS	PLI
RDBSFMT	PLI
RDBSIVRT	PLI
RDBSLCT	PLI
RDBSTAT	PLI
RDBSTRIP	PLI
RDBSTRT	PLI
RDBTIE	ASM
RDBTSSIO	ASM
RDBUPDST	PLI
RDBUSER	PLI
RDBVERBS	PLI
RDBVS	ASM
RDBWRIT	PLI
RDBXPND	PLI
RINDEX	ASM
RMTTSUP	ASM
RSOURCE	ASM
RTIMERS	ASM
RTSATTN	ASM
RTSPRC	ASM
RTSSTRT	ASM
RTSTESTX	ASM
RTSUPER	ASM
RTS11X10	ASM
RUSERID	ASM
STATIC	TSS
TRNSCT	TSS
TRNSCT#	TSS
USERJCIN	TSS
UTILITYS	NON